We are a team of ~~nerds~~ PhDs with a solid math/cs background, specialized in blockchain and zero-knowledge proof cryptography

check out some of our faces

get in touch!

What have we been doing?

- Efficient implementation of large integers.
- Finite fields *(Montgomery form, CIOS, Binary GCD algorithm)*
- Elliptic curves *(Affine, projective, Jacobian coordinates, Mixed addition. MSM, Tate pairing, Optimal ate pairing)*
- Polynomial commitment schemes *(KZG, FRI + Merkle trees, IPA)*
- Proving systems *(Pinocchio, Plonk, STARKs)*
- Library for building and evaluating Plonk circuits.
- Circuit programming *(Noir, Plonky2)*
- Prover compatibility: Lambdaworks and other libraries through reverse engineering *(Starkware's Stone prover, Winterfell)*
- Solidity contract programming *(cryptographic primitives and protocols)*
- Protocol audits *(Stark)*
- MSM optimization *(GLV, endomorphisms, Pippenger)*
- FFT implementation *(and extensions like Circle STARKs)*
- Circuits for generating proofs of ECDSA and ECIES execution
- Formal specification of STARKs protocol
- ACIR / ACVM backend development

Some publicly available write-ups we authored:

- STARKs protocol
- An overview of the Stone Cairo STARK Prover

Programming Languages: anything Turing complete (or less) =P

But usually, for:

- Performance *(Rust, Go, C++, Crystal)*
- Proof of concept *(Python, Sage)*
- Domain-specific *(Solidity, Yul, Noir)*